REMARKS

This Amendment responds to the Office Action mailed on August 15, 2008. Claim 25 is canceled and claim 24 is amended to include the limitations of now-canceled claim 25 in order to place the present application in better condition for appeal. Reconsideration of the remaining claims is respectfully requested in light of the above amendments.

The rejections of independent claims 31, 34, and 40 and dependent claim 25 are improper because the Examiner has failed to establish that the cited Flowers reference anticipates each element of the claims. The Final Office Action repeatedly makes the following citation to Flowers as corresponding to numerous claim limitations, without even attempting to explain how these limitations are taught by any of the cited parts of Flowers:

> abstract, column 2, lines 50-67 & column 3, lines 1-50 & column 4, lines 50-67 & column 5, lines 1-25 & fig 2, including the explanation provided in the Independent claim

As shown above, the Examiner has essentially cited to the entire Flowers specification in general for support and cannot cite to any specific points for support because they are clearly not present. The Applicant therefore submits that the Final Office Action fails to show correspondence between the Flowers reference and each element of the rejected independent claims, and thus fails to make out a *prima facie* showing of anticipation under 35 U.S.C. § 102(b).

Furthermore, the Examiner could not have made a *prima facie* case under § 102(b) because the cited Flowers reference clearly does not disclose a method, performed by a server, of facilitating the processing of font data for electronic transfers to client devices, that includes the limitations recited in independent claims 24, 31, 34, and 40 and dependent claim 25.

Claim 24, as now amended to include the limitations of dependent claim 25, recites a method with steps performed by a server including "transferring the lacking font structure data and the text data to the designated device, wherein the client device permanently stores the received font structure data in the client device." The cited Flowers reference fails to disclose or otherwise suggest anything close to permanently storing font data in a client device. Rather, the Flowers reference specifically teaches away from storing font data structure on the client device:

> [T]he FAF font server eliminates the need for font storage and/or font rendering processors in all clients, including printer clients. (Flowers, Col. 12, ll. 62-64).

The client in Flowers only stores the table of font names in order to more efficiently communicate with the font server. (Flowers, Col. 3, ll. 39-45). The Examiner's rejection of claim 25 is thus clearly erroneous as it fails to point to any relevant portion of Flowers to support the rejection, and the portions cited to teach away from the present application.

Claim 31 recites a method with steps performed by a server including transferring to the test device, and operatively refraining from transferring the font structure data to the device in response to determining, in step iii, that the device already has the font data structure." If the client device already has stored the font data, the server does not have to send augment font data. In contrast, in the Flowers reference the FAF server must always send font data to the client device because the client device does not store any font data. (Flowers, Col. 12, ll. 62-64 and Fig. 2). Therefore, because the client device in Flowers does not store any font data, the FAF server does not make a comparison of stored font data on a client device, and refrain from sending font data that the client device already has stored. The Examiner clearly erred in construing the teachings of Flowers, because if the FAF font server refrains from sending the font data, the device in Flowers would be inoperable. Thus, the Examiner has committed clear error in reading Flowers to disclose the exact opposite of what it actually teaches.

Claim 34 recites a method with steps performed by a server including "determining whether another font identifier exists that is the same as said not found identifier." The Flowers reference does not disclose anything remotely close to finding the same missing font data if the data is not available on the client device. Flowers discloses that if the identified font is not supported by the client device (i.e., the client does not have a license to use the font), then Flowers translates the font into another type that is supported by the client device. (Flowers, Col. 7, ll. 12-18). In no manner does Flowers attempt to locate font data not supported by the client device, and it was clear error for the Examiner to read Flowers as disclosing such.

Claim 40 recites a method with steps performed by a server including "transferring both the text data and the font structure data for said not found font identifier to the designated device." As recited above regarding claim 34, Flowers does not send any font data for missing font structures that are not supported by the client device. Instead, in Flowers, the font is translated to another font that can be supported by the client device. Therefore, Flowers cannot be read to teach a transfer of text data and font structure data when the corresponding font identifier is not found on the client device. Flowers cannot, and does not, send font structure data to a client device when it is not supported by the device. The Examiner has again committed clear error in reading Flowers to teach what is clearly not present.

Conclusion

For at least the above reasons, the applicant respectfully submits that the pending claims are patentable over the cited references and are in condition for allowance.

Respectfully submitted,

JONES DAY

Joseph M. Sauer (Reg. No. 47,919)
Jones Day
North Point, 901 Lakeside Avenue
Cleveland, Ohio 44114
(216) 586-7506